



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2000010766 A**(43) Date of publication of application: **14.01.00**(51) Int. Cl. **G06F 9/06**(21) Application number: **10171360**(71) Applicant: **FUJITSU LTD**(22) Date of filing: **18.06.98**(72) Inventor: **AKIYAMA TOSHIHIKO**

(54) **SYSTEM APPLYING PROGRAM CORRECTION  
EXTRACTION AND STORAGE MEDIUM  
STORING PROGRAM APPLYING CORRECTION  
EXTRACTION**

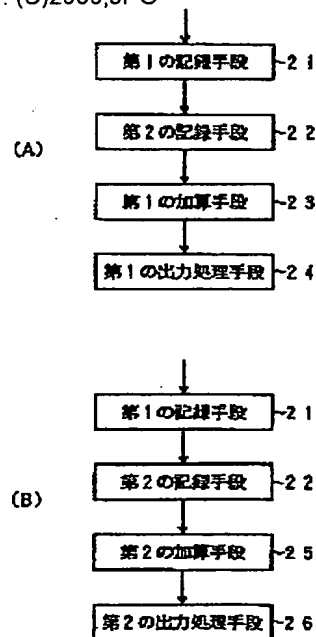
the correction number of the program correction are outputted to a second storage means if the total frequency in running is equal to or larger than the preliminarily determined prescribed frequency.

(57) Abstract:

COPYRIGHT: (C)2000,JPO

**PROBLEM TO BE SOLVED:** To reduce the man-hour of examination for extraction of program correction having an influence upon a client and to enable a user to immediately understand that there is a possibility of the occurrence of a fault.

**SOLUTION:** A module name given by a prescribed system of a program which is executed at the time of system operation and the frequency in running are recorded in a control table by a first recording means 21, and they are taken out from the control table and are recorded in a first storage means by a second recording means 22 at the time of stop of the system, and a first addition means 23 obtains frequencies in running of modules having the same function name as the object module name of program correction from the control table and the first storage means in the unit of functions and adds them at the time of receiving the program correction, and the obtained total frequency in running and a preliminarily determined prescribed frequency are compared with each other by a first output processing means 24, and a first degree of danger and



**THIS PAGE BLANK (USPTO)**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-10766  
(P2000-10766A)

(43) 公開日 平成12年1月14日 (2000.1.14)

(51) Int.Cl.<sup>7</sup>  
G 0 6 F 9/06

識別記号  
4 1 0  
5 4 0

F I  
G 0 6 F 9/06

テマコート\* (参考)  
4 1 0 H 5 B 0 7 6  
5 4 0 U  
5 4 0 A

審査請求 未請求 請求項の数 7 O L (全 19 頁)

(21) 出願番号 特願平10-171360

(22) 出願日 平成10年6月18日 (1998.6.18)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 秋山 寿彦

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74) 代理人 100093584

弁理士 宮内 佐一郎 (外1名)

Fターム(参考) 5B076 EA02

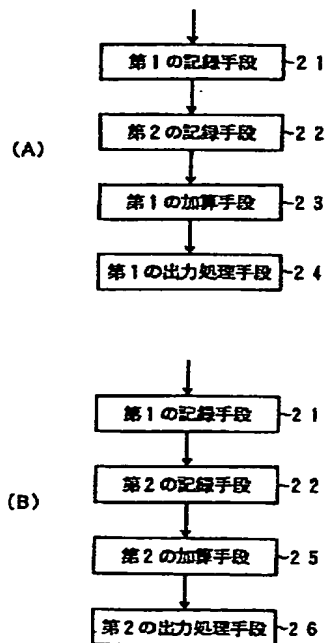
(54) 【発明の名称】 プログラム修正抽出適用システムおよび修正抽出適用プログラムを格納した記憶媒体

(57) 【要約】

【課題】 顧客に影響のあるプログラム修正の抽出の調査工数を削減し、障害が発生する可能性のあることを利用者にすぐに分かるようにする。

【解決手段】 第1の記録手段21によりシステムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録し、第2の記録手段22によりシステムの停止時に制御表から取り出して第1の記憶手段にモジュール名と走行回数を記録し、第1の加算手段23がプログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を機能単位で制御表および第1の記憶手段から求めて加算し、第1の出力処理手段24により加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第1の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理する。

本発明の原理説明図



## 【特許請求の範囲】

【請求項 1】 計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、

システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第 1 の記録手段と、

システムの停止時に前記制御表から取り出して第 1 の記憶手段にモジュール名と走行回数を記録する第 2 の記録手段と、

プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を機能単位で前記制御表および前記第 1 の記憶手段から求めて加算する第 1 の加算手段と、

加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第 1 の危険度とプログラム修正の修正番号を第 2 の記憶手段に出力処理する第 1 の出力処理手段と、を備えたことを特徴とするプログラム修正抽出適用システム。

【請求項 2】 計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、

システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第 1 の記録手段と、

システムの停止時に前記制御表から取り出して第 1 の記憶手段にモジュール名と走行回数を記録する第 2 の記録手段と、

プログラム修正を受信したとき求めた対象モジュールの走行回数の合計をモジュール単位で前記制御表および前記第 1 の記憶手段から求めて加算する第 2 の加算手段と、

加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第 2 の危険度とプログラム修正の修正番号を第 2 の記憶手段に出力処理する第 2 の出力処理手段と、を備えたことを特徴とするプログラム修正抽出適用システム。

【請求項 3】 請求項 1、2 記載のプログラム修正抽出適用システムにおいて、

前記第 1 の危険度および第 2 の危険度が指定されたまたは前記第 2 の危険度のみが指定されたかを判別して前記第 2 の記憶手段より該当する危険度の修正番号を読み出す読出手段と、

読み出された修正番号のプログラム修正を修正適用のため受信メモリより取り出す取出手段と、を備えたことを特徴とするプログラム修正抽出適用システム。

【請求項 4】 計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、

プログラム修正を受信したとき、自動的に前記請求項 2

および請求項 3 を連続して実施するかまたは前記請求項 1、請求項 2 および請求項 3 を連続して実施して、修正適用プログラムを自動的に起動することを特徴とするプログラム修正抽出適用システム。

【請求項 5】 計算機システムの保守において供給されるプログラム修正を抽出して適用するための修正抽出適用プログラムを格納した記憶媒体において、

システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第 1 の記録手段と、

システムの停止時に前記制御表から取り出して第 1 の記憶手段にモジュール名と走行回数を記録する第 2 の記録手段と、

プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を前記制御表および前記第 1 の記憶手段から求めて加算する第 1 の加算手段と、

加算した総走行回数と予め定めた所定の回数とを比較して第 1 の危険度とプログラム修正の修正番号を第 2 の記憶手段に出力処理する第 1 の出力処理手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体。

【請求項 6】 計算機システムの保守において供給されるプログラム修正を抽出して適用するための修正抽出適用プログラムを格納した記憶媒体において、

システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第 1 の記録手段と、

システムの停止時に前記制御表から取り出して第 1 の記憶手段にモジュール名と走行回数を記録する第 2 の記録手段と、

プログラム修正を受信したとき求めた対象モジュールの走行回数の合計を前記制御表および前記第 1 の記憶手段から求めて加算する第 2 の加算手段と、

加算した総走行回数と予め定めた所定の回数とを比較して第 2 の危険度とプログラム修正の修正番号を第 2 の記憶手段に出力処理する第 2 の出力処理手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体。

【請求項 7】 請求項 5、6 記載の修正抽出適用プログラムを格納した記憶媒体において、

前記第 1 の危険度および第 2 の危険度が指定されたまたは前記第 2 の危険度のみが指定されたかを判別して前記第 2 の記憶手段より該当する危険度の修正番号を読み出す読出手段と、

読み出された修正番号のプログラム修正を修正適用のため受信メモリより取り出す取出手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体。

【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、保守センタなどから送られてくるプログラム修正を抽出して適用するプログラム修正抽出適用システムおよびプログラム修正を抽出して適用するための修正抽出適用プログラムを格納した記憶媒体に関する。ホストコンピュータの保守の考え方は、一度システムを決めて運用してしまったらそのシステムは凍結して、数年間更新しないのが普通である。これは、ホストコンピュータの場合は、障害が発生すると業務がストップしてしまい影響が大きいために、安定しているシステムのままで運用したためである。

【0002】一方、OSを含むアプリケーションプログラムのソフトウェア障害の修正は、メーカー側から定期的に出荷されており、その中にはすぐに適用しないと顧客システムに影響を及ぼすものも含まれているため、定期的に適用を推進するようにしている。この場合、顧客に影響を与えるプログラム修正の抽出、適用を手間や時間をかけずに行うことが要望されている。

## 【0003】

【従来の技術】従来、保守センタより送られてきたプログラム修正から顧客システムに影響のあるプログラム修正を抽出するときは、できるだけ顧客システムの更新を少なくするために定期的に発行されるプログラム修正の説明票を見て、実際に顧客に影響しそうな緊急なプログラム修正を抽出し、個別に適用している。

【0004】このようなプログラム修正の内容の説明票としては、例えば、図25および図26に示すようなものがある。これらの説明票には、図25および図26に示すようにシステム名、製品名、製品世代、コンポーネント名、プログラム番号、修正番号、適用条件が記載され、障害内容、原因、処置、代行方法が説明されている。

【0005】ところが、顧客に影響のあるプログラム修正かを調べるには、顧客がどの製品のどの機能を使用しているかを知っている必要がある。また、プログラム修正の出荷数は、製品数の増加に伴って増加するため、調べるだけでも工数がかかってしまう。この作業は全て手作業で行っていた。

## 【0006】

【発明が解決しようとする課題】このような従来のプログラム修正の抽出適用方法にあつては、顧客に影響のあるプログラム修正の抽出を手作業で行っていたため、プログラム修正の抽出のための調査の工数がかかるという問題があった。また、プログラム修正を適用する操作作業も手間がかかっていた。

【0007】本発明は、このような従来の問題に鑑みてなされたものであつて、調査の工数がかからず、また、適用の操作作業も手間がかからないプログラム修正抽出適用システムおよび修正抽出適用プログラムを格納した記憶媒体を提供することを目的とする。

## 【0008】

【課題を解決するための手段】この目的を達成するため、本発明は図1のように構成する。請求項1の発明は、計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第1の記録手段21と、システムの停止時に前記制御表から取り出して第1の記憶手段にモジュール名と走行回数を記録する第2の記録手段22と、プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を機能単位で前記制御表および前記第1の記憶手段から求めて加算する第1の加算手段23と、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第1の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理する第1の出力処理手段24と、を備える。

【0009】請求項2の発明は、計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第1の記録手段22と、システムの停止時に前記制御表から取り出して第1の記憶手段にモジュール名と走行回数を記録する第2の記録手段23と、プログラム修正を受信したとき求めた対象モジュールの走行回数の合計をモジュール単位で前記制御表および前記第1の記憶手段から求めて加算する第2の加算手段25と、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第2の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理する第2の出力処理手段26と、を備える。

【0010】請求項3の発明は、請求項1、2記載のプログラム修正抽出適用システムにおいて、前記第1の危険度および第2の危険度が指定されたまたは前記第2の危険度のみが指定されたかを判別して前記第2の記憶手段より該当する危険度の修正番号を読み出す読出手段と、読み出された修正番号のプログラム修正を修正適用のため受信メモリより取り出す取出手段と、を備えたことを特徴とするプログラム修正抽出適用システム。

【0011】請求項4の発明は、計算機システムの保守において供給されるプログラム修正を抽出して適用するプログラム修正抽出適用システムにおいて、プログラム修正を受信したとき、自動的に前記請求項2および請求項3を連続して実施するかまたは前記請求項1、請求項2および請求項3を連続して実施して、修正適用プログラムを自動的に起動する。

【0012】請求項5の発明は、計算機システムの保守において供給されるプログラム修正を抽出して適用する

ための修正抽出適用プログラムを格納した記憶媒体において、システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第1の記録手段と、システムの停止時に前記制御表から取り出して第1の記憶手段にモジュール名と走行回数を記録する第2の記録手段と、プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を前記制御表および前記第1の記憶手段から求めて加算する第1の加算手段と、加算した総走行回数と予め定めた所定の回数とを比較して第1の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理する第1の出力処理手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体よりなる。

【0013】請求項6の発明は、計算機システムの保守において供給されるプログラム修正を抽出して適用するための修正抽出適用プログラムを格納した記憶媒体において、システムの運転時に実行するプログラムの所定の体系で命名されたモジュール名と走行回数を制御表に記録する第1の記録手段と、システムの停止時に前記制御表から取り出して第1の記憶手段にモジュール名と走行回数を記録する第2の記録手段と、プログラム修正を受信したとき求めた対象モジュールの走行回数の合計を前記制御表および前記第1の記憶手段から求めて加算する第2の加算手段と、加算した総走行回数と予め定めた所定の回数とを比較して第2の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理する第2の出力処理手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体よりなる。

【0014】請求項7の発明は、請求項5、6記載の修正抽出適用プログラムを格納した記憶媒体において、前記第1の危険度および第2の危険度が指定されたまたは前記第2の危険度のみが指定されたかを判別して前記第2の記憶手段より該当する危険度の修正番号を読み出す読出手段と、読み出された修正番号のプログラム修正を修正適用のため受信メモリより取り出す取出手段と、を備えたことを特徴とする修正抽出適用プログラムを格納した記憶媒体よりなる。

【0015】このような構成を備えた本発明によれば、プログラムの所定の体系で命名されたモジュール名と走行回数を記録しておき、プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を機能単位で制御表および第1の記憶手段から求めて加算し、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第1の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理するため、顧客に影響のあるプログラム修正の抽出の調査工数を削減することができ、送られてきたプログラム修正を適用しないと障害が発生する可能性があることが利用者にすぐ分かる。

【0016】また、プログラムの所定の体系で命名されたモジュール名と走行回数を記録しておき、プログラム修正を受信したとき求めた対象モジュールの走行回数の合計をモジュール単位で制御表および第1の記憶手段から求めて加算し、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第2の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理するため、顧客に影響のあるプログラム修正の抽出の調査工数を削減することができ、送られてきたプログラム修正を一刻も早く適用しないと、非常に危険であることが利用者にすぐ分かる。

【0017】また、第1の危険度および第2の危険度が指定されたまたは第2の危険度のみが指定されたかを判別して第2の記憶手段より該当する危険度の修正番号を読み出し、読み出した修正番号のプログラム修正を修正適用のため受信メモリより取り出すため、適用が必要なプログラム修正のみを簡単な操作で適用することができる。

【0018】さらに、プログラム修正を受信したとき、自動的に請求項2および請求項3を連続して実施するかまたは請求項1、請求項2および請求項3を連続して実施して、修正適用プログラムを自動的に起動するため、利用者が意識することなく、顧客に必要なプログラム修正のみを全て自動的に適用することができる。

【0019】

【発明の実施の形態】図2は本発明の一実施形態を示す全体構成図である。図2において、1はプログラム修正抽出適用システム2の本体であり、本体1には入力部3、表示部4、プリンタ5、ディスク6、保守センタ7、磁気媒体（テープ）8などがそれぞれ接続されている。ディスク6内にはアプリケーションプログラムである複数のモジュール9が格納され、本体1内に送られる。モジュール9はROM10内に格納されたOS11の制御下でCPU12により走行する。

【0020】モジュール9のモジュール名は、図3に示すように、機能単位に体系的に命名されている。モジュール名（A、B、C・・・Z）は、小分類、中分類、大分類により構成される。A、B、C・・・Zの文字数は任意に構成される。例えば、モジュール名の例を図4に示す。図4に示すように、3文字目までが大分類、4文字目までが中分類、5文字目までが小分類を構成し、8文字目までがモジュール名を示す。図4の場合、5文字目（小分類）までが同一の機能を表わしている。例えば、モジュール名「JYMGIL62」、「JYMGINSN」、「JYMGISNA」のうち「JYMG I」が同一の機能を表わしている。

【0021】図2に戻って、本体1のRAM13内には制御表14が展開される。モジュール9の走行回数を記録するため、OS11がモジュール9に制御を渡す時点でモジュール名と走行回数を制御表14に記録する。制

御表14は、図5に示すように、検索し易いようにモジュール名の昇順になるように配置する。例えば、制御表14のABC、ABCD、ABCDE、ABCDEFはモジュール名を示し、昇順になっている。

【0022】図6に示すように、制御表14中にすでにモジュール名が存在する場合は、新たにモジュール名を作成せずに、走行回数のみ+1カウントアップする。例えば、モジュール名「ABCDE I J K」はすでに存在しているので、走行回数を+1カウントアップして2回とする。また、モジュール名が制御表14に存在しないときは、図7に示すように、モジュール名を制御表14の内部に展開し、走行回数を1回とする。例えば、モジュール名「ABCDEF」が制御表14に存在しないときは、モジュール名「ABCDEF」を作成し、走行回数を1回として、次ポイントで次のモジュール名「ABCDE I J K」に渡す。

【0023】制御表14の情報は、保存が必要だが、システム運用中は、処理速度に影響を与えないために、保存せず、システム停止時にまとめて第1の記憶手段としてのVSAMメモリ15に保存する。VSAMメモリ15は、モジュール名をキーに持つキー順データセットメモリである。この時も、既に同名のモジュールのレコードが存在する場合は、走行回数を制御表14中の走行回数分カウントアップして更新し、同名のレコードが存在しない場合は、新たにレコードを挿入する。

【0024】例えば、図6に示すように、システム停止時には、制御表14からモジュール名と走行回数をVSAMメモリ15に記録する。例えばレコードとしてモジュール名「ABCDEFGH」走行回数40回、モジュール名「ABCDE I J K」走行回数50回が記録される。再び図2に戻って、保守センタ7から定期的に送られてくるプログラム修正は、受信メモリ16内に格納される。保守センタ7からだけではなく、磁気媒体（テープ）8で入手したプログラム修正も受信メモリ16内に格納される。パラメータメモリ17には、パラメータとして所定の走行回数や顧客に影響を与える第1の危険度b、第2の危険度aが入力部3より設定される。顧客に影響のあるプログラム修正を機能単位で求めるとき、例えば走行回数10回が設定され、顧客に影響のあるプログラム修正をモジュール単位で求めるとき、例えば走行回数30回が設定される。

【0025】また、顧客に影響を与えるパラメータとしての危険度aおよび危険度bの適用が指定されるかまたは危険度aのみの適用が指定される。危険度bは、顧客に影響のあるプログラム修正かを機能単位で求めるとき、VSAMメモリ15と制御表14の走行回数の加算値がパラメータメモリ17の所定の走行回数以上るとき、判定され、第2の記憶手段としての順データセットメモリ18に格納されるとともに、表示部4に表示され、必要によりプリンタ5に出力される。

【0026】また、危険度aは、顧客に影響のあるプログラム修正かをモジュール単位で求めるとき、VSAMメモリ15と制御表14の走行回数の加算値がパラメータメモリ17の所定の回数以上るとき、判定され、順データセットメモリ18に格納されるとともに表示部4に表示され、必要によりプリンタ5に出力される。ディスク6内には、本発明のプログラムA～D19および修正適用プログラム20が格納されている。

【0027】プログラムAは、図8に示すように、第1の記録手段としての第1の記録部21と第2の記録手段としての第2の記録部22を有する。第1の記録部21は、システムの運転時に実行するプログラムのモジュール名と走行回数を制御表14に記録する。第2の記録部22は、システムの停止時に制御表14から取り出して第1の記憶手段としてのVSAMメモリ15にモジュール名と走行回数を記録する。

【0028】プログラムBは、図9に示すように、第1の加算手段としての第1の加算部23と第1の出力処理手段としての第1の出力処理部24を有する。第1の加算部23は、プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を制御表14およびVSAMメモリ15から求めて加算する。

【0029】第1の出力処理部24は加算した総走行回数とパラメータ17に格納された予め定めた所定の回数とを比較して、所定の回数より多いとき、第1の危険度としての危険度bとプログラム修正の修正番号を第2の記憶手段としての順データセットメモリ18に出力処理し、また、表示部4に危険度bを表示する。例えば、図10に示すように、保守センタ7より修正番号が「A001」の対象モジュール「ABCDEFGH」を受信したとき、VSAMメモリ15には同一機能名を有するモジュール「ABCDEFGH」の走行回数が40回、「ABCDE I J K」の走行回数の50回が格納されていると、加算した総走行回数は、パラメータメモリ17の所定の回数10回より多いので、危険度bが判定され、危険度bは表示部4に表示され、順データセットメモリ18に格納される。

【0030】プログラムCは、図11に示すように、第2の加算手段としての第2の加算部25と第2の出力処理手段としての第2の出力処理部26を有する。第2の加算部25はプログラム修正を受信したとき求めた対象モジュールの走行回数の合計を制御表14およびVSAMメモリ15から求めて加算する。第2の出力処理部26は加算した総走行回数とパラメータメモリ17に格納された予め定めた所定の回数とを比較して、所定の回数以上るとき第2の危険度としての危険度aとプログラム修正の修正番号を順データセットメモリ18に出力処理し、また表示部4に危険度aを表示する。

【0031】例えば、図12に示すように、保守センタ

7から修正番号「A001」の対象モジュール「ABCDEFGHIJ」を受信したとき、VSAMメモリ15には同一モジュール「ABCDEFGHIJ」の走行回数40回が格納されており、総走行回数は、パラメータメモリ17に格納されている所定の回数30回を超えるので、危険度aと判定し、危険度aと修正番号「A001」を表示部4に表示し、また、順データセットメモリ18に格納する。

【0032】プログラムDは、図13に示すように、読出手段としての読出部27および取出手段としての取出手段28を有する。読出部27は第1の危険度bおよび第2の危険度aが指定されたまたは第2の危険度aのみが指定されたかをパラメータメモリ17の設定により判別して順データセットメモリ18より該当する危険度aまたはbの修正番号を読み出す。取出手段28は読み出された修正番号のプログラム修正を修正適用のため受信メモリより取り出す。

【0033】図14に示すように、例えば、パラメータメモリ17内に危険度aおよび危険度bの適用が設定されたときは、危険度a、bに該当する修正番号「A001」、「A003」、「A002」を順データセットメモリ18から読み出し、受信メモリ16から修正番号のモジュール「ABCDEFGHIJ」を取り出し、修正適用プログラム20をOS11の制御下で起動する。

【0034】図15はプログラムAの動作を説明するフローチャートである。図15において、まず、ステップS1でシステムが停止したか否かを判別し、システムの通常運転時には、ステップS2に進み、システムが停止したときは、ステップS5に進む。ステップS2でOS11が制御を渡すモジュール9を把握する。

【0035】次に、ステップS3で制御表14にモジュール名と走行回数を記録する。すでに制御表14にモジュール名があるときは、走行回数を+1とする。制御表14にないときは、モジュール名と走行回数1回を記録する。そして、ステップS4でOS11がモジュール9に制御を渡し、ステップS1に戻る。例えば、図19に示すように、通常運転時「ARSORRCV」というモジュール9が呼び出されたとする。制御表14の「ARSORRCV」に走行回数1回をカウントしようとするが、すでに制御表14に「ARSORRCV」が存在するため（システムを起動してから1回走行しているため）、走行回数のカウントアップで $1+1=2$ 回と記録する。また、「ARSORRCV」の走行前にはモジュール「ARSORSND」は20回、モジュール「ARSORSSND」は1回が記録されている。

【0036】ステップS1でシステムの停止時には、ステップS5で制御表14の1つのモジュールを取り出す。そして、ステップS6でそのモジュール名と走行回数をVSAMメモリ15に記録する。すでにモジュール名がVSAMメモリ15にある場合には走行回数を加算

する。VSAMメモリ15にない場合にはモジュール名と走行回数よりなるレコードを追加する。次に、ステップS7で次ポイントがあるか判別し、あるときは、ステップS5に戻り、ないときは終了とする。

【0037】例えば、図19に示すように、システムの停止時にVSAMメモリ15にはARSORRCV 1回、ARSORSND 50回、ARSORSSND 100回が記録される。このように、モジュール9の走行回数を常にカウントすることで、どのくらいその機能またはそのモジュールを使用しているかわかり、顧客に影響のあるプログラム修正を抽出する目安となる。

【0038】図16はプログラムBの動作を説明するフローチャートである。図16において、まず、ステップS11で保守センタ7からのプログラム修正の受信を契機にプログラムBを起動し、パラメータメモリ17に指定された走行回数を読み込む。例えば、図19に示すように、パラメータメモリ17で指定された走行回数は50回であるとする、この走行回数を読み込む。次に、ステップS12でプログラム修正の受信により対象モジュール名を求め、「機能名」に該当するモジュール9を求める。図19に示すように、保守センタ7より送られてきた修正番号「A001」の対象モジュールが「ARSORRCV」であるとする、前提条件としてモジュール名の先頭から5文字までが機能を表わす文字であるので、「ARSOR」が機能名であり、走行回数を求めるモジュール9は「ARSORRCV」と「ARSORSND」になる。

【0039】次に、ステップS13で同一の「機能名」をもつモジュール9の走行回数の合計を制御表14より求める。図20に示すように、「ARSORRCV」は制御表14の走行回数が2回、「ARSORSND」は制御表14の走行回数が20回であり、合計の走行回数は22回となる。次に、ステップS14で同一の「機能名」をもつモジュール9の走行回数の合計とVSAMメモリ15より求めて、ステップS13の結果に加算する。図20に示すように、「ARSORRCV」のVSAMメモリ15の走行回数は1回、「ARSORSND」のVSAMメモリ15の走行回数は50回であるので、合計の走行回数は51回となり、制御表14の合計走行回数22回に加算すると総走行回数は73回となる。

【0040】次に、ステップS15でプログラム修正を受信する受信メモリ16内に他の対象モジュールがあるか判別し、あるときはステップS12に戻り、ないときはステップS16に進む。ステップS16では総走行回数とパラメータメモリ17の所定の回数とを比較する。図20に示すように、総走行回数は73回であり、パラメータメモリ17の走行回数は50回であり、総走行回数が多いので、ステップS17に進む。

【0041】なお、パラメータ17の走行回数の方が多



いときは、ステップS17に進まないで、処理を終了とする。ステップS17では表示部4に修正番号(A001)と「危険度b」を出力し、順データセットメモリ18に修正番号(A001)と「危険度b」を記録する(図19、参照)。この場合には、プログラム修正の抽出の調査工数を削減することができ、保守センタ7から送られてきたプログラム修正を適用しないと、障害が発生する可能性があることが利用者にすぐに分る。

【0042】図17はプログラムCの動作を説明するフローチャートである。図17において、まず、ステップS21で保守センタ7からのプログラム修正の受信を契機にプログラムBを起動し、パラメータメモリ17で指定された回数を読み込む。例えば、図19に示すように、パラメータメモリ17で指定された走行回数は、10回であるとする、これを読み込む。次に、ステップS22でプログラム修正の受信により対象モジュール名を求める。図19に示すように、保守センタ7より送られてきた修正番号「A001」の対象モジュールが「ARSORRCV」であり、これからモジュール名を求める。

【0043】次に、ステップS23で対象モジュールの走行回数を制御表14より求める。図21に示すように、「ARSORRCV」の制御表14の走行回数は2回である。次に、ステップS24で対象モジュールの走行回数をVSAMメモリ15より求めて、ステップS23の結果に加算する。図21に示すように、「ARSORRCV」のVSAMメモリ15の走行回数は1回であり、制御表14の走行回数2回に加算すると総走行回数は3回となる。

【0044】次に、ステップS25でプログラム修正を受信する受信メモリ16内に他の対象モジュールがあるか判別し、あるときはステップS22に戻り、ないときはステップS26に進む。ステップS26では総走行回数とパラメータメモリ17の所定の回数とを比較する。図21に示すように、例えば総走行回数は3回であり、パラメータメモリ17の走行回数は10回のときは、パラメータメモリ17の走行回数の方が多いので、この場合には処理を終了とする。

【0045】一方、総走行回数の方が多いときは、ステップS27に進む。ステップS27では表示部4に修正番号(A001)と「危険度a」を出力し、順データセットメモリ18に修正番号(A001)と「危険度a」を記録する(図19、参照)。このように、プログラム修正の抽出の調査工数を削減することができ、また、危険度aを記録したときは、保守センタ7から送られてきたプログラム修正を一刻も早く適用しないと非常に危険であることが利用者にすぐ分る。

【0046】また、プログラムBでは危険度bが記録され、プログラムCでは危険度aが記録されないときは、危険度bのままであり、修正対象の機能を使っているの

で、適用が必要であるが、頻繁には発生しないということになる。図18はプログラムDの動作を説明するフローチャートである。図18において、まず、ステップS31でパラメータメモリ17で指定された内容を読み込む。例えば、図19に示すように、危険度aおよび危険度bの適用が指定されていれば、これを読み込む。また、危険度aのみの適用が指定されていれば、これを読み込む。

【0047】次に、ステップS32でパラメータメモリで「危険度a」が指定されたか判別し、指定されたときは、ステップS33で順データセットメモリ18より「危険度a」の修正番号を全て読み込む。「危険度a」が指定されていないときは、ステップS34でパラメータメモリ17で「危険度b」が指定されたか判別し、「危険度b」が指定されたときは、ステップS35で順データセットメモリ18より「危険度b」の修正番号を全て読み込む。なお、モジュール名が重複した場合には1つにまとめる。図19に示すように、「危険度b」が指定されたときは、順データセットメモリ18より「危険度b」の修正番号「A001」を読み込む。「危険度b」が指定されないときは、ステップS36に進む。

【0048】次に、ステップS36では読み込ませたプログラム修正を受信メモリ16から取り出す。「危険度b」の修正番号「A001」を読み込んだときは、その修正番号のモジュール「ARSORRCV」を受信メモリ16から取り出し、プログラムDが終了したとき、修正適用プログラム20を起動する。すなわち、危険度aおよび危険度bの適用を指定したときは、修正番号「A001」は危険度bであるため、適用される(図19、参照)。なお、「危険度a」のみの適用を指定したときは、修正番号「A001」は危険度bであるため、適用されない。このように、顧客に影響のあるプログラム修正のみを簡単な操作で適用することができる。

【0049】図22は本発明の他の実施形態を示す全体構成図である。図22において、ディスク6内には自動起動プログラム29が設けられ、顧客に影響のあるプログラム修正をオペレータの確認なしで全て自動的に適用するようにしている。自動起動プログラム29は、プログラムA、プログラムB、プログラムC、修正適用プログラム20を順次自動的に起動するかまたはプログラムA、プログラムC、修正適用プログラム20を順次自動的に起動する。

【0050】パラメータメモリ17には顧客に影響のあるプログラム修正を機能単位で求めるための走行回数、顧客に影響のあるプログラム修正をモジュール単位で求めるための走行回数が予め設定されている。また、パラメータメモリ17には「危険度aおよび危険度b」の適用、または「危険度a」の適用が予め指定されている。

【0051】図23は自動起動プログラム29の動作を説明するフローチャートである。図23において、ま

ず、ステップS41で保守センタ7からのプログラム修正の受信を契機に一連のジョブを起動する。ステップS42ではプログラムAを起動する。プログラムAではモジュールaの走行回数を常にカウントすることでどのくらいその機能またはモジュールを使用しているか確認することができる。

【0052】次に、ステップS43でプログラムCを起動する。プログラムCでは顧客に影響のあるプログラム修正をモジュール単位で求めるために、所定の走行回数以上のとき危険度aを記録する。この場合には、保守センタ7から送られてきたプログラム修正を一刻も早く適用しないと非常に危険であることが分かる。次に、ステップS44でプログラムDを起動する。プログラムDでは危険度aに該当する修正番号から適用が必要なプログラム修正のみを取り出す。

【0053】次に、ステップS45で修正適用プログラム20を起動する。こうして、プログラム修正が一刻も早く適用しないと非常に危険であるモジュールを抽出し、オペレータが意識することなく、顧客が必要なプログラム修正のみを自動的に適用することができる。図24は自動起動プログラム29の他の動作を説明するフローチャートである。

【0054】図24において、図23のステップS42とステップS43との間にステップS42Aが追加して設けられている。その他の構成は、図23と同様である。プログラムBでは、顧客に影響のあるプログラム修正を機能単位で求めるために、所定の走行回数以上のとき、危険度bを記録する。図24においては、保守センタ7からのプログラム修正を受信したとき、プログラムA、プログラムB、プログラムC、プログラムDおよび修正適用プログラム20を順次自動的に起動する。

【0055】この場合には、障害が発生する可能性のあるプログラム修正もオペレータの確認なしに自動的に抽出され、適用される。なお、保守センタ7からのプログラム修正に限らず、磁気媒体（テープ）8からプログラム修正を入力しても良い。

【0056】

【発明の効果】以上説明してきたように、本発明によれば、プログラムの所定の体系で命名されたモジュール名と走行回数を記録しておき、プログラム修正を受信したときその対象モジュール名より同一の機能名をもつモジュールの走行回数の合計を機能単位で制御表および第1の記憶手段から求めて加算し、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第1の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理するため、顧客に影響のあるプログラム修正の抽出の調査工数を削減することができ、送られてきたプログラム修正を適用しないと、障害が発生する可能性があることが利用者にすぐ分かる。

【0057】また、プログラムの所定の体系で命名され

たモジュール名と走行回数を記録しておき、プログラム修正を受信したとき求めた対象モジュールの走行回数の合計をモジュール単位で制御表および第1の記憶手段から求めて加算し、加算した総走行回数と予め定めた所定の回数とを比較して所定の走行回数以上のとき第2の危険度とプログラム修正の修正番号を第2の記憶手段に出力処理するため、顧客に影響のあるプログラム修正の抽出の調査工数を削減することができ、送られてきたプログラム修正を一刻も早く適用しないと、非常に危険であることが利用者にすぐ分かる。

【0058】また、第1の危険度および第2の危険度が指定されたまたは第2の危険度のみが指定されたかを判別して第2の記憶手段より該当する危険度の修正番号を読み出し、読み出した修正番号のプログラム修正を修正適用のため受信メモリより取り出すため、適用が必要なプログラム修正のみを簡単な操作で適用することができる。

【0059】さらに、プログラム修正を受信したとき、自動的に請求項2および請求項3を連続して実施するかまたは請求項1、請求項2および請求項3を連続して実施して、修正適用プログラムを自動的に起動するため、利用者が意識することなく、顧客に必要なプログラム修正のみを全て自動的に適用することができる。

【図面の簡単な説明】

【図1】本発明の原理説明図

【図2】本発明の実施形態を示す全体構成図

【図3】モジュール名の説明図

【図4】モジュール名の例を示す図

【図5】制御表の構成例を示す図

【図6】モジュールの走行回数の記録を説明する図

【図7】制御表の展開を示す図

【図8】プログラムAの構成例を示す図

【図9】プログラムBの構成例を示す図

【図10】プログラムBの説明図

【図11】プログラムCの構成例を示す図

【図12】プログラムCの説明図

【図13】プログラムDの構成例を示す図

【図14】プログラムDの説明図

【図15】プログラムAの動作を説明するフローチャート

【図16】プログラムBの動作を説明するフローチャート

【図17】プログラムCの動作を説明するフローチャート

【図18】プログラムDの動作を説明するフローチャート

【図19】プログラム修正の抽出、適用の説明図

【図20】走行回数の総計を示す図（その一）

【図21】走行回数の総計を示す図（その二）

【図22】本発明の他の実施形態を示す全体構成図

【図23】自動起動プログラムの動作を説明するフローチャート（その一）

【図24】自動起動プログラムの動作を説明するフローチャート（その二）

【図25】従来のプログラム修正内容説明票の例を示す図（その一）

【図26】従来のプログラム修正内容説明票の例を示す図（その二）

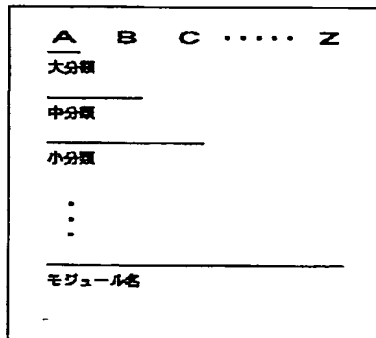
【符号の説明】

- 1：本体
- 2：プログラム修正抽出適用システム
- 3：入力部
- 4：表示部
- 5：プリンタ
- 6：ディスク
- 7：保守センタ
- 8：磁気媒体（テープ）
- 9：モジュール
- 10：ROM

- 11：OS
- 12：CPU
- 13：RAM
- 14：制御表
- 15：VSAMメモリ（第1の記憶手段）
- 16：受信メモリ
- 17：パラメータメモリ
- 18：順データセットメモリ（第2の記憶手段）
- 19：プログラムA～D
- 20：修正適用プログラム
- 21：第1の記録部（第1の記録手段）
- 22：第2の記録部（第2の記録手段）
- 23：第1の加算部（第1の加算手段）
- 24：第1の出力処理部（第1の出力処理手段）
- 25：第2の加算部（第2の加算手段）
- 26：第2の出力処理部（第2の出力処理手段）
- 27：読出部（読出手段）
- 28：取出部（取出手段）
- 29：自動起動プログラム

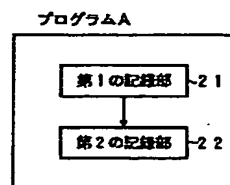
【図3】

モジュール名の説明図



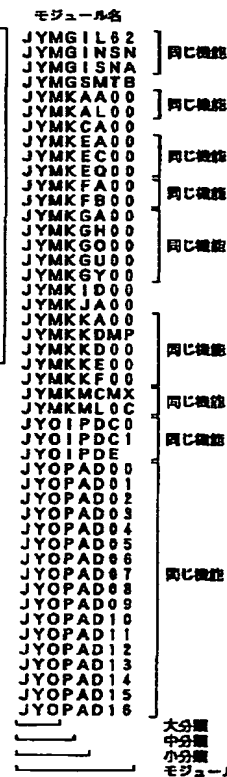
【図8】

プログラムAの構成例を示す図



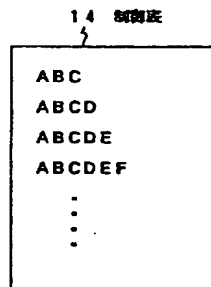
【図4】

モジュール名の例を示す図



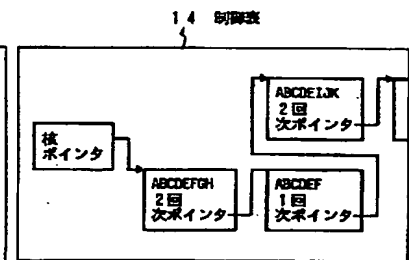
【図5】

制御表の構成例を示す図



【図7】

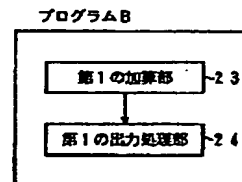
制御表の展開を示す図



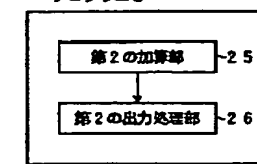
【図11】

【図9】

プログラムBの構成例を示す図



プログラムCの構成例を示す図



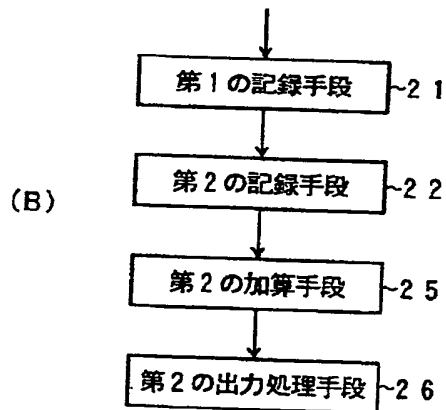
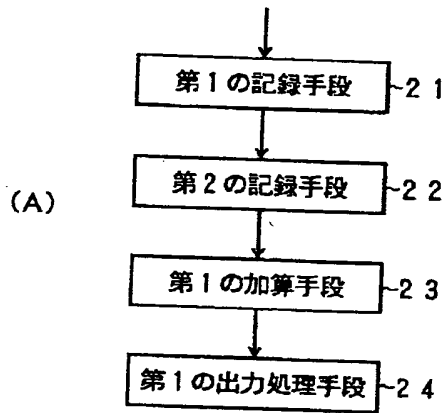
【図20】

走行回数の統計を示す図（その一）

モジュール名	制御表	VSAMメモリ
ARSORCV	2回	1回
ARSORSND	20回	50回
総計		73回

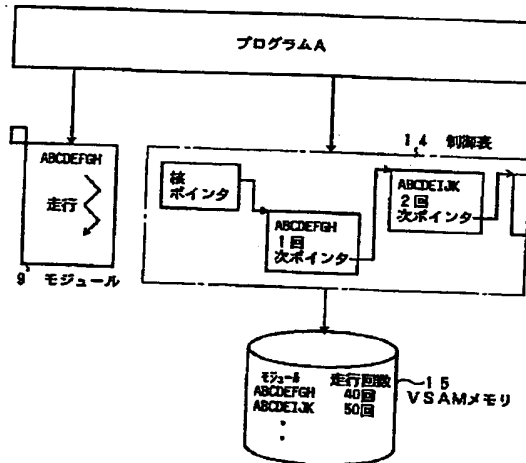
【図1】

## 本発明の原理説明図



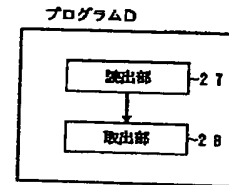
【図6】

## モジュールの走行回数の記録を説明する図



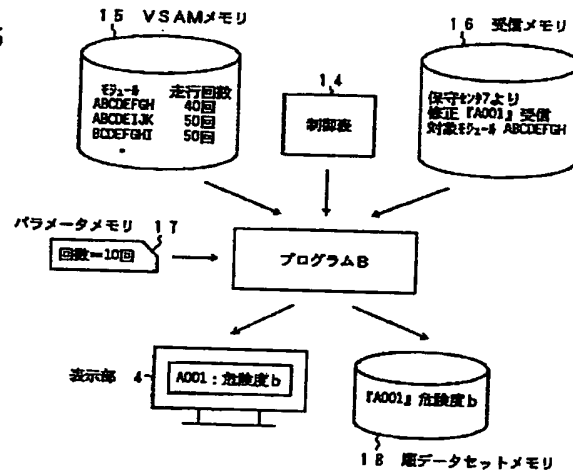
【図13】

## プログラムDの構成例を示す図



【図10】

## プログラムBの説明図



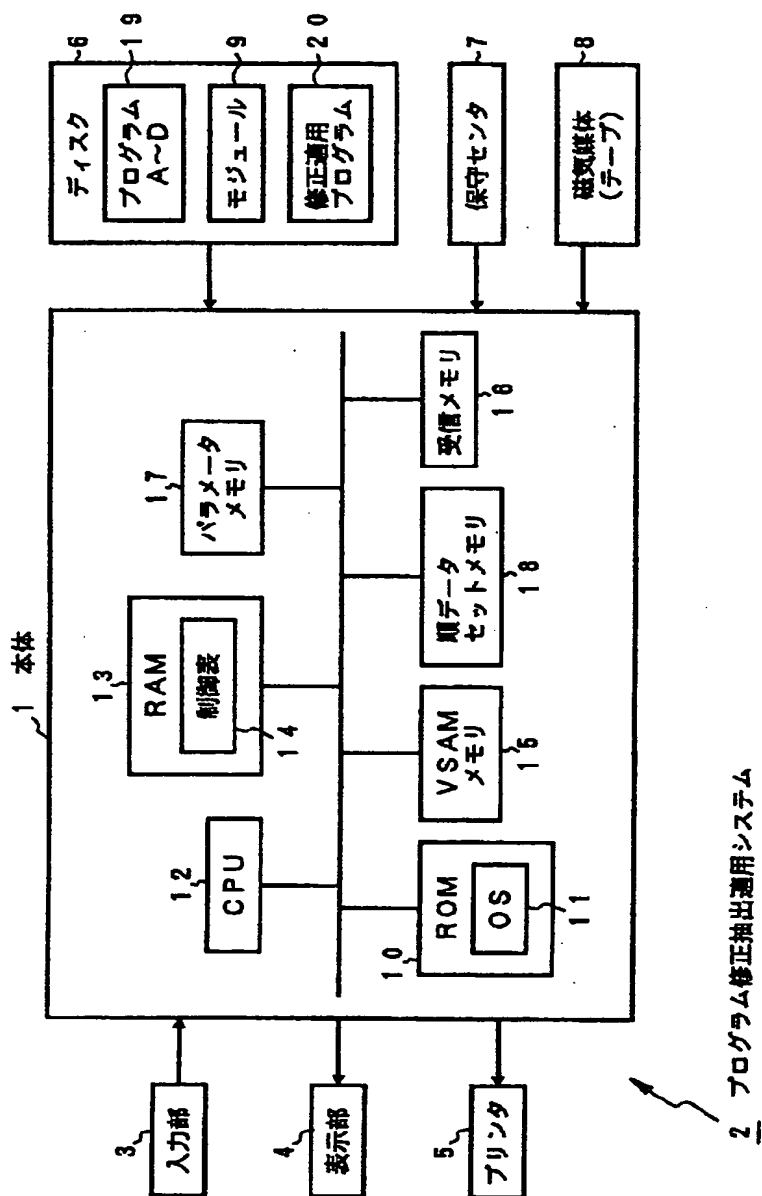
【図21】

## 走行回数の総計を示す図 (その二)

モジュール名	制御表	VSAMメモリ
ARSORRCV	2回	1回
総計	3回	

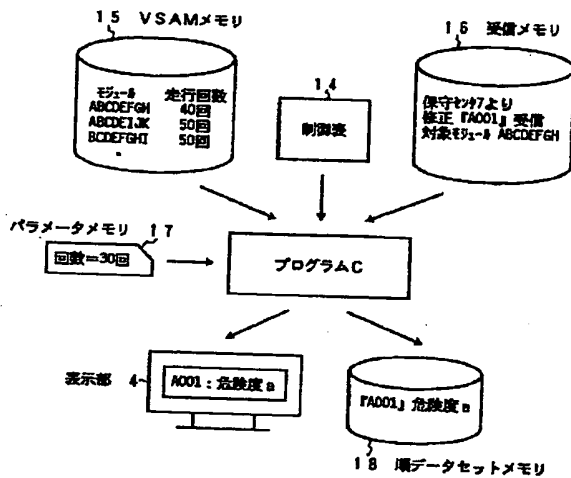
【図2】

本発明の実施形態を示す全体構成図



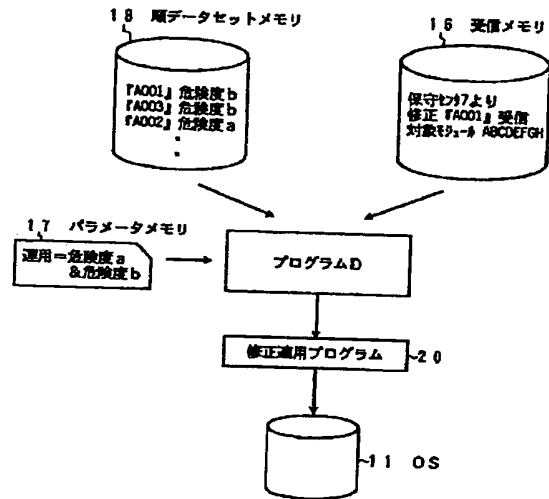
【図12】

プログラムCの説明図



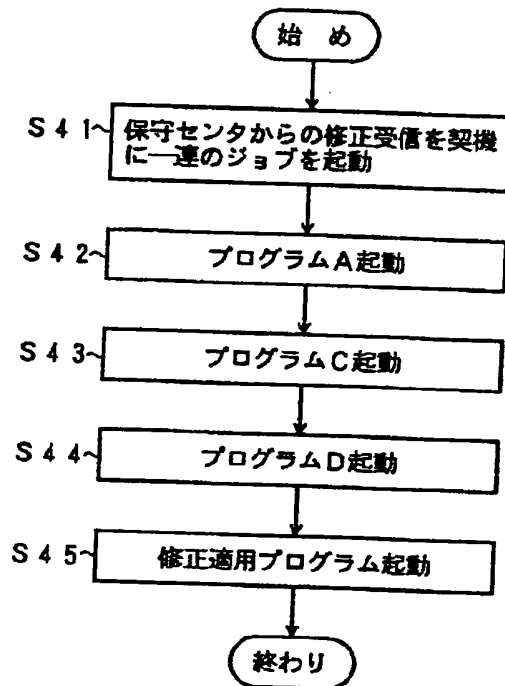
【図14】

プログラムDの説明図



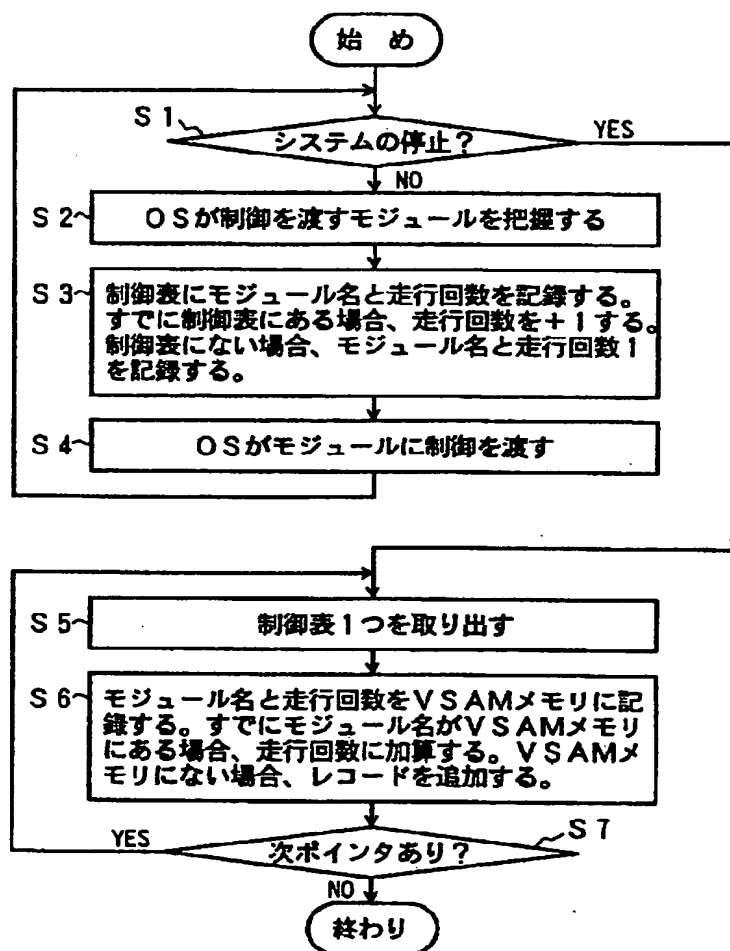
【図23】

自動起動プログラムの動作を説明するフローチャート (その一)



【図15】

## プログラムAの動作を説明するフローチャート



【図26】

## 従来のプログラム修正内容説明票の例を示す図 (その二)

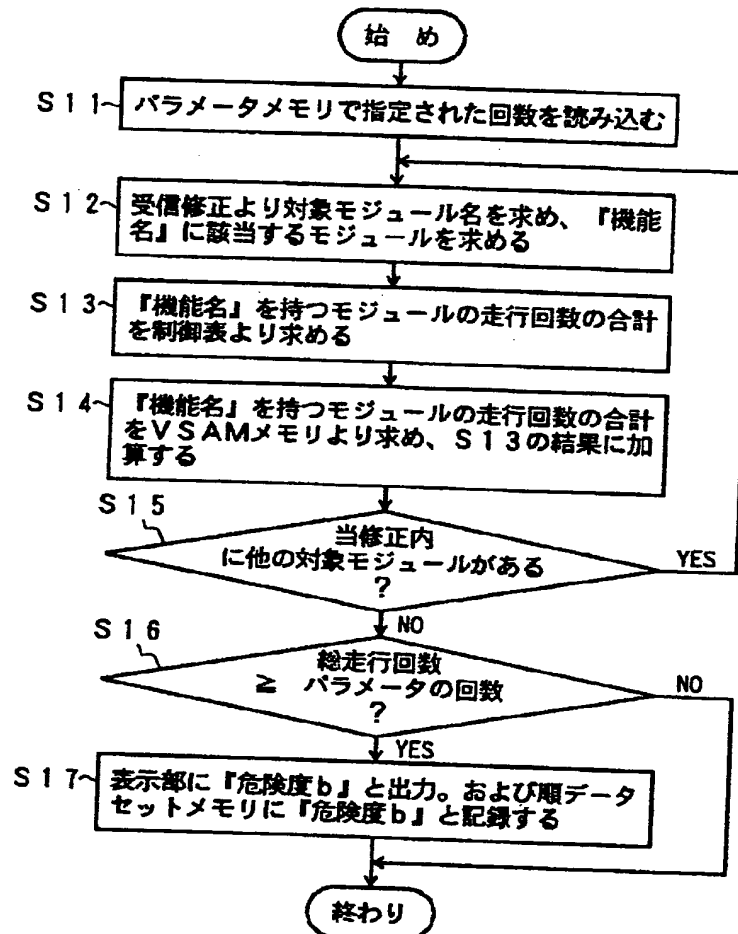
## プログラム修正内容説明票

システム名 : OS4 P番号 : P111112  
 製品名 : SEI HIN2 修正番号 : AV11112  
 製品世代 : V10L10  
 コンポーネント名 : COMPO2  
 適用条件 : V91071-V97121

1. 障害内容  
以下の条件のとき、ダウンロード済オーバーレイを使用したWRTの出力形式と、WRTの直接出力による出力形式が異なる。  
 1) オーレイパターンをダウンロードした場合  
 2) オーレイパターンのサイズがA4ランド・B4ランド・A5ポート・B5ポート以外のパターンの場合
2. 原因  
オーバーレイパターンの判定処理に誤りがあったため
3. 処置  
オーバーレイパターンの判定処理を修正する
4. 代行方法  
なし

【図16】

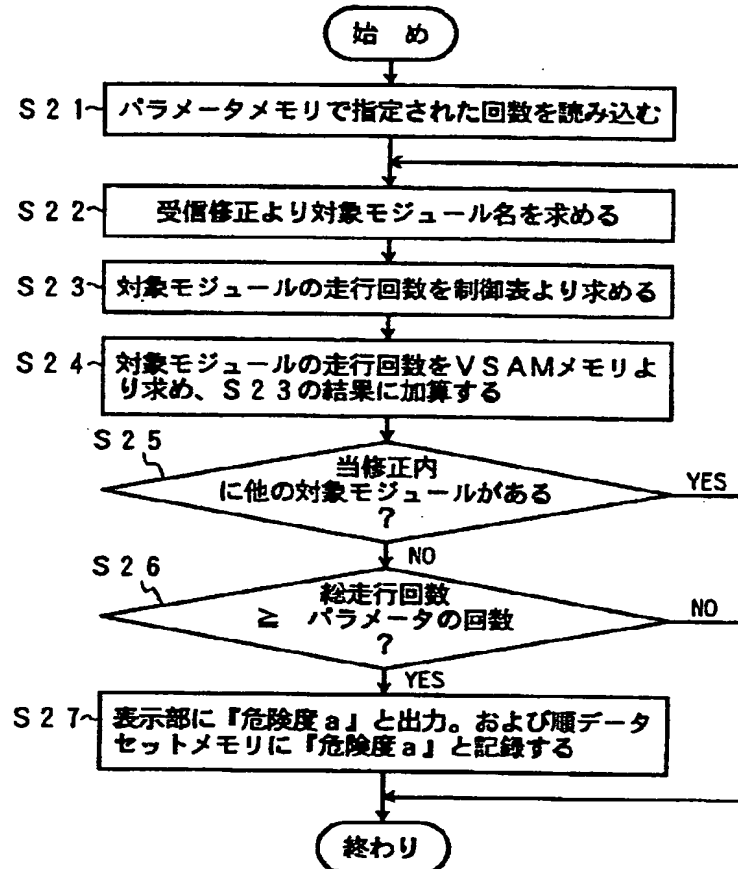
プログラムBの動作を説明するフローチャート





【図17】

## プログラムCの動作を説明するフローチャート



【図25】

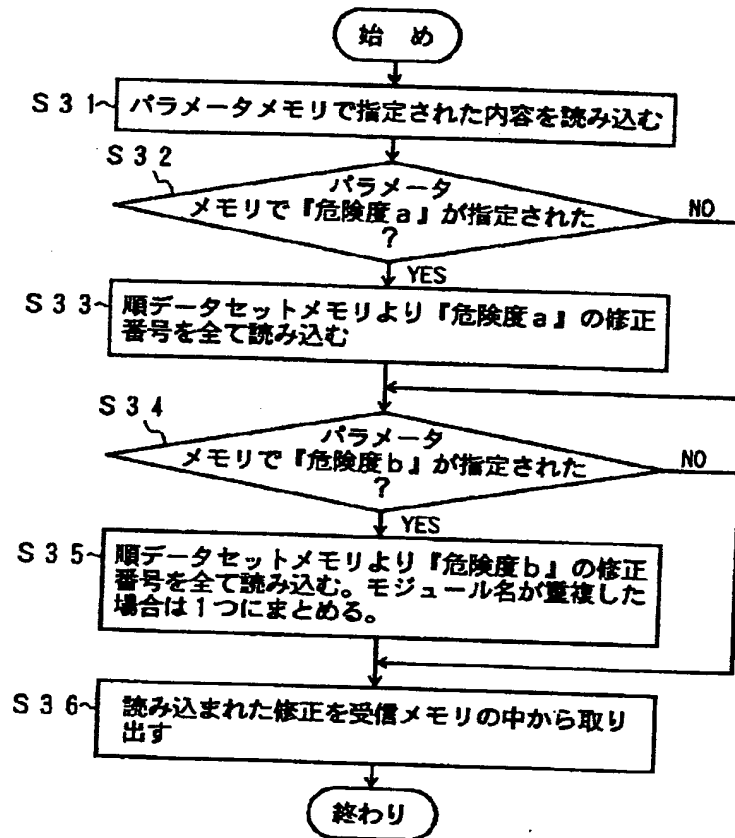
従来のプログラム修正内容説明票の例を示す図（その一）

プログラム修正内容説明票

システム名	: OS4	P番号	: P111111
製品名	: SEIHINI	修正番号	: AV11111
製品世代	: V10L10		
コンポーネント名	: COMP01		
適用条件: V00000-V97121			
1. 障害内容 MP構成のシステムの場合、システムがコード53-304でダウンすることがある。当該障害は以下の条件で発生し易い傾向がある。 1) KEITA11または、KEITA12のシステムである。 2) CPUを共用部で使用する。 3) WORKボリュームに対するI/O負荷が高い。 (I/O実行待ち発生する)			
2. 原因 ワークボリューム制御の処理で異常が発生した場合のリカバリ処理に誤りがあった			
3. 処置 ワークボリューム制御のリカバリ処理の誤りを修正する			
4. 代行方法 なし			

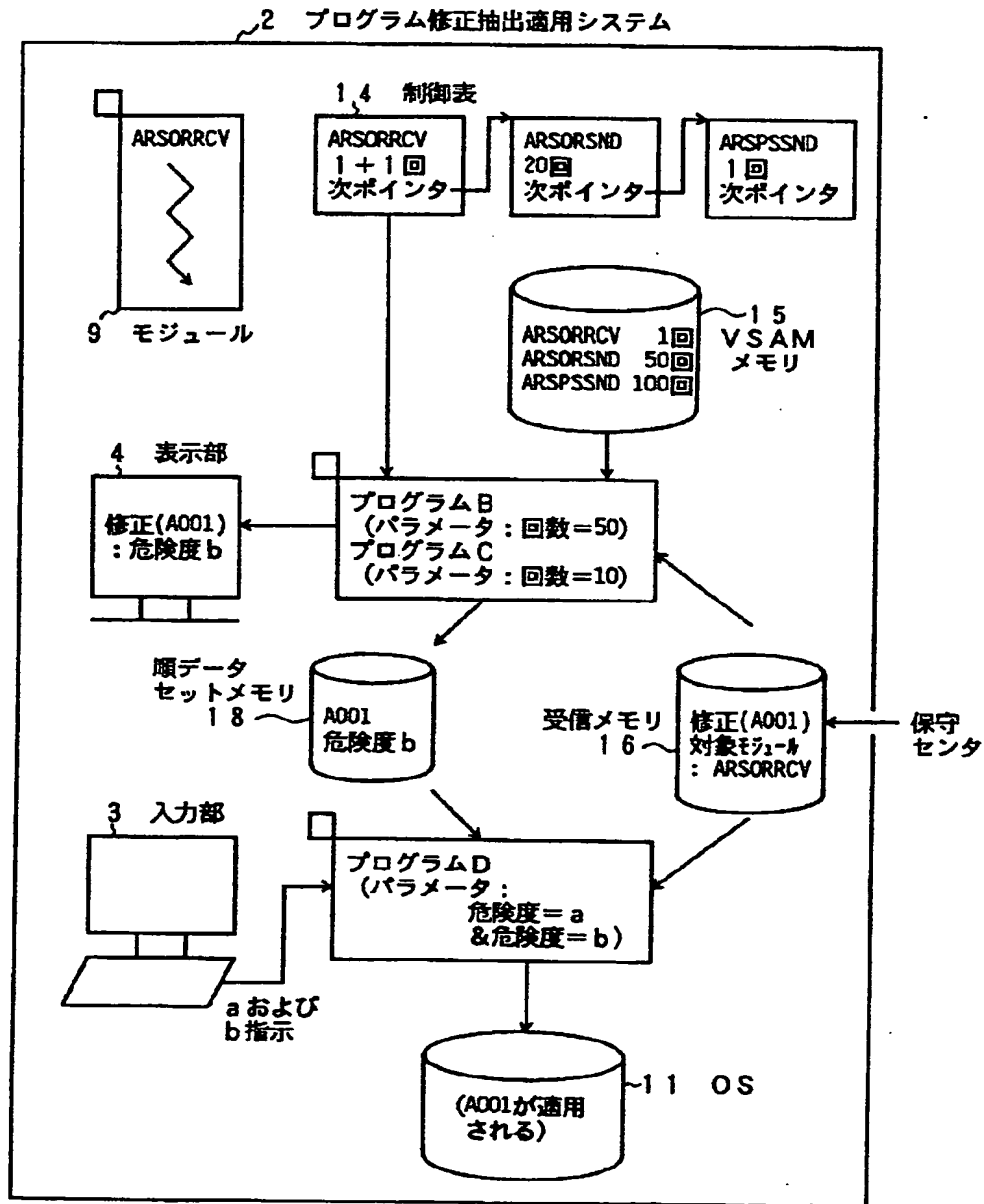
【図18】

プログラムDの動作を説明するフローチャート



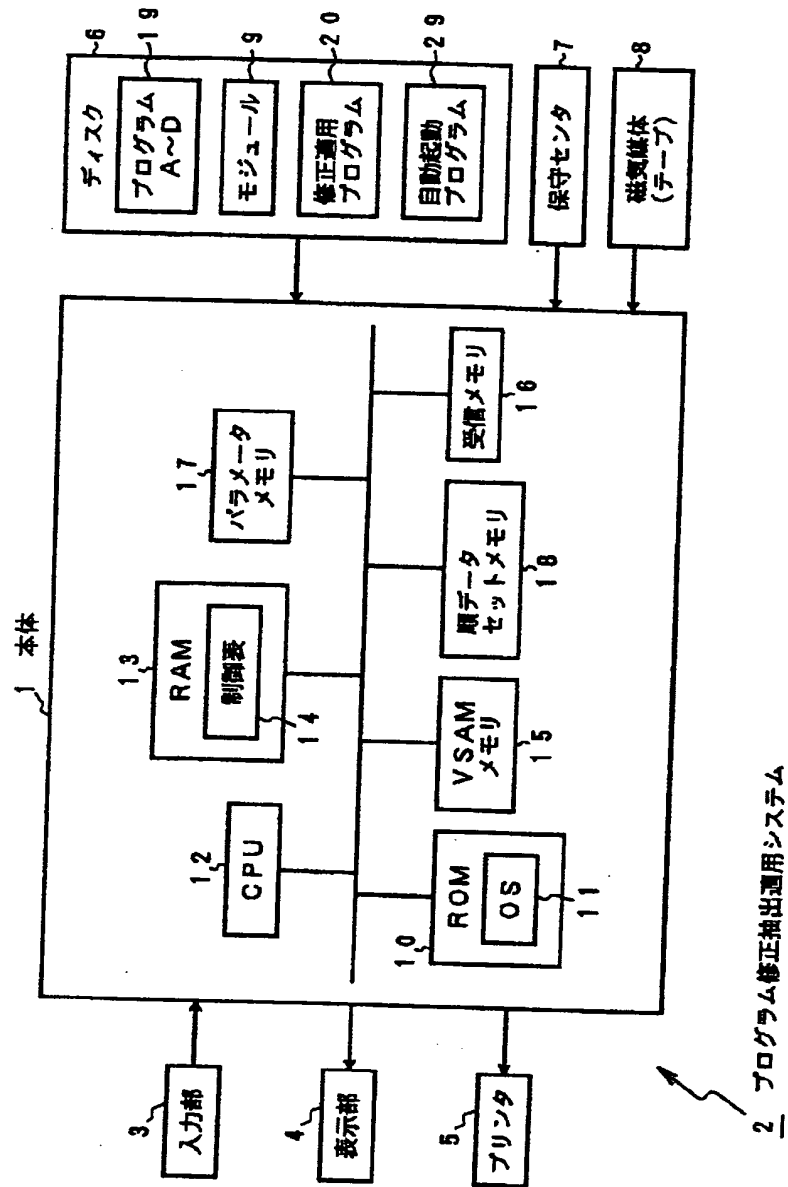
【図19】

## プログラム修正の抽出、適用の説明図



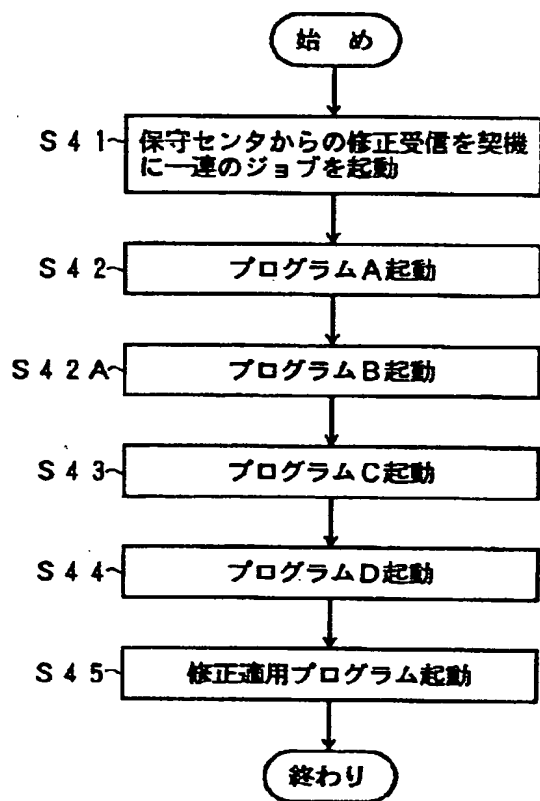
【図22】

本発明の他の実施形態を示す全体構成図



【図24】

自動起動プログラムの動作を説明するフローチャート（その二）



**THIS PAGE BLANK (USPTO)**